DMPR-PS: A NOVEL APPROACH FOR PARKING-SLOT DETECTION USING DIRECTIONAL MARKING-POINT REGRESSION

Junhao Huang[1], *Lin Zhang*[1,*], *Ying Shen*[1,*], *Huijuan Zhang*[1], *Shengjie Zhao*[1], *Yukai Yang*[2]

[1]School of Software Engineering, Tongji University, Shanghai, China
[2]Department of Statistics, Uppsala University, Uppsala, Sweden

ABSTRACT

The self-parking system plays an important role in autonomous driving, and one of its critical issues is parking-slot detection. Previous studies in this field are mostly based on off-the-shelf models designed for universal purposes, which have various limitations in solving specific problems. In this paper, we propose a parking-slot detection method using directional marking-point regression, namely DMPR-PS. Instead of utilizing multiple off-the-shelf models, DMPR-PS uses a novel CNN-based model specially designed for directional marking-point regression. Given a surround-view image I, the model predicts position, shape and orientation of each marking-point on I. From marking-points, parking-slots on I could be easily inferred using geometric rules. DMPR-PS outperforms state-of-the-art competitors on the benchmark dataset with a precision rate of 99.42% and a recall rate of 99.37%, while achieving a real-time detection speed of 12ms per frame on Nvidia Titan Xp. To make the results reproducible, the source code is available at https://github.com/Teoge/DMPR-PS.

***Index Terms*—** Self-parking system, parking-slot detection, deep convolutional neural network

1. INTRODUCTION

Being the last step in autonomous driving, the self-parking system is an important part of an unmanned driving system and has attracted the interest of a large number of researchers. In developing such a system, one of the key issues to be addressed is how to correctly detect and locate parking-slots with in-vehicle sensors. Solutions to this problem fall into two categories, free-space-based ones and vision-based ones.

Free-space-based approaches designate a target parking position by recognizing a sufficient vacant space between adjacent vehicles. Various kinds of range-finding sensors have been used in these approaches, such as ultrasonic sensors [1, 2], laser scanners [3], short-range radars [4, 5], etc. However, free-space-based approaches have an inherent drawback that they rely on vehicles that have already been properly parked as references. In other words, these approaches fail to work in an open area with no vehicles nearby. In order to overcome this drawback, more and more researchers turn their attention to the vision-based approaches, expecting to find a more universal and robust solution.

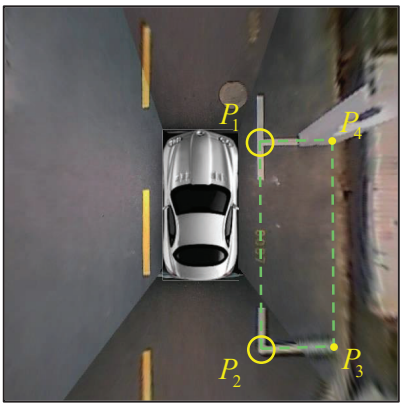


Fig. 1. An example of parking-slot markings.

Instead of recognizing a free space, a vision-based approach detects parking-slot-markings to locate parking-slots. Parking-slot-markings are the line-shaped markings painted on the ground to indicate a valid parking space. Thus, detecting parking-slot-markings to locate a parking-slot is more legitimate than recognizing a free space and better conforms to human drivers' perception. For these reasons, the study of vision-based parking-slot detection has drawn a lot of attention from researchers and is also the focus of this paper.

Fig. 1 shows an example of parking-slot-markings in a surround-view image. These markings indicate a rectangular parking space, whose vertices are P_1, P_2, P_3 and P_4. Among four conceptual lines that form this rectangle, line P_1P_2 is called an "entrance-line". Line P_1P_4 and P_2P_3 are called "separating-lines". Marking-points are defined as the junction points of entrance-lines and separating-lines, such as P_1 and P_2. Assuming the lengths of separating-lines are known beforehand as a priori knowledge, the parking-slot detection problem can be formulated as the problem of detecting an ordered marking-point pair (P_1, P_2), whose order is defined as the anticlockwise order of the four vertices.

*Corresponding author. Email: {cslinzhang, yingshen}@tongji.edu.cn

978-1-5386-9552-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICME.2019.00045

As described above, the detection target of the parking-slot detection problem is different from the one of common object detection problems. Thus, existing object detection frameworks could not be applied directly to solve this problem. Most existing solutions to this problem decompose it into sub-problems and solve them separately. Representative approaches in this field are briefly reviewed as follows.

1.1. Related work

Among those vision-based parking-slot detection approaches, some of them, *e.g.* [6, 7], require manual operations on the interactive interface to detect a certain parking-slot. The apparent drawback of these approaches is that they are not fully automated. The fully automated approaches can be categorized into line-based ones and point-based ones.

Line-based approaches [8, 9, 10, 11, 12, 13, 14, 15] are based on marking-line detection. Usually, they first detect edges of marking-lines in the image, and then use line fitting algorithm to predict the equations of marking-lines. When marking-lines are ready, their geometric relations are analyzed to recognize entrance-lines and separating-lines. Then parking-slots in the image can be located. A variety of methods have been exploited to detect the edges of marking-lines, including Sobel filter [9, 11, 14], segmentation neural network [8, 16], Canny edge detector [10], Line Segment Detector [15] and cone-hat filter [12]. They also use various line fitting algorithms, such as Hough Transform [9, 11], Radon Transform [10, 16], RANSAC (RANdom SAmpling Consensus) [13, 14], and customized line clustering algorithms [12].

Point-based approaches [17, 18, 19, 20] are based on detection of marking-points. They first detect marking-points in the image, then for each pair of marking-points, different methods are used to determine whether it forms a parking-slot entrance-line and which orientation the parking-slot is. Suhr and Jung [17, 18] used the Harris corner detector to detect corners of marking-points and used the template matching technique to determine the marking-points' shapes and orientations. Then, locations of parking-slots were inferred based on above information. Li *et al.* [19] resorted to a boosting decision tree to detect marking-points and then applied Gaussian line filters to find parking-slots' entrance-lines and determine their orientations. Quite recently, Zhang *et al.* proposed a CNN (convolutional neural networks) based approach, namely DeepPS [20]. It first uses a CNN to detect marking-points and then uses another CNN to classify local image patterns determined by marking-point pairs. Zhang *et al.*'s experiments [20] indicate that DeepPS outperforms all low-level feature based methods and also corroborate the effectiveness of CNN for solving this issue.

1.2. Our motivations and contributions

As aforementioned, the recently proposed CNN-based approach DeepPS [20] is a state-of-the-art solution for vision-based parking-slot detection. However, though DeepPS is proven to be both effective and efficient, it decomposes the parking-slot detection problem into two problems in computer vision, and resorts to two different CNNs to solve them individually. Yet given the strong correlation between these two problems, the feature extraction procedures of these two CNNs are highly repetitive. Consequently, the efficiency of DeepPS can be further improved.

If we manage to obtain information required for parking-slot inference using a single CNN, the resulting parking-slot detection scheme would be much more efficient. Inspired by such an idea, in this work, we propose a novel and highly efficient parking-slot detection approach based on directional marking-point regression, which is referred to as DMPR-PS for short.

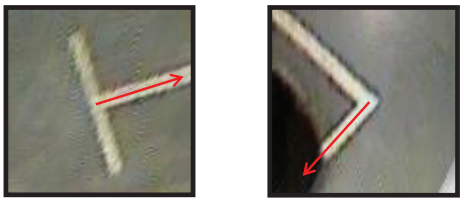
In this approach, we first introduce a new concept named "directional marking-point", which can be used to accurately describe the marking-line pattern around a marking-point. Then based on this concept, a novel CNN-based multi-attribute regression model is proposed to detect all directional marking-points in the image. By detecting directional marking-points, we manage to obtain sufficient marking-line information required for parking-slot inference. Thus we are able to directly infer the locations of all parking-slots in the image from directional marking-points.

Our regression model, namely Directional Marking-Point Regression model (DMPR), predicts all attributes of directional marking-points simultaneously. During the prediction, all directional marking-points of a given image are predicted in a single forward evaluation. The efficacy and efficiency of DMPR-PS have been corroborated in experiments conducted on the public benchmark dataset ps2.0 [20].

2. DIRECTIONAL MARKING-POINT

A directional marking-point is actually a local image pattern characterized by a marking-point and its neighborhood. It has three attributes, position, shape and orientation.

Position. As above-mentioned, marking-points are the junction points of marking-lines. However, strictly speaking, marking-lines are line-shaped markings with certain width. Two intersecting marking-lines form a squared junction area.



(a) T-shaped (b) L-shaped

Fig. 2. Two kinds of marking-point patterns with red arrows indicating their orientations.

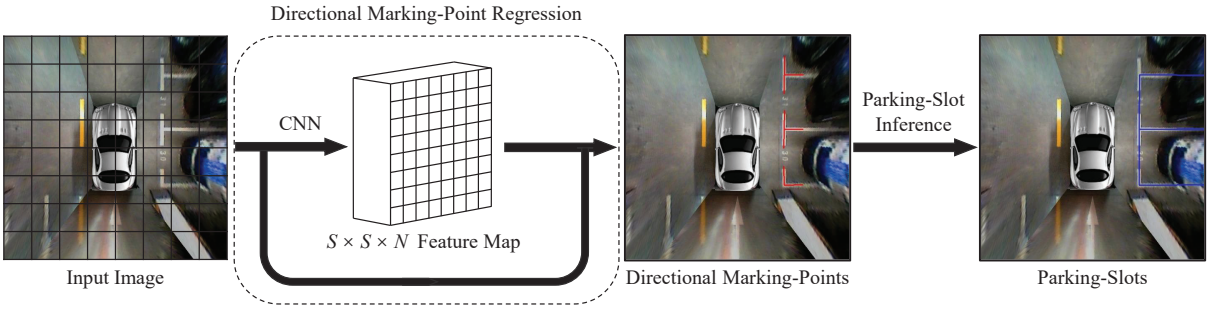


Fig. 3. The flowchart of our proposed method DMPR-PS. It comprises two steps, Directional Marking-Point Regression and Parking-Slot Inference. First, a CNN-based regression model is used to detect directional marking-points in the surround-view image. Then, parking-slots in the image are inferred from detected marking-points according to their geometric relations.

Here we define the position of a directional marking-point as the center of this junction area.

Shape. According to the shape of the marking-point pattern, all kinds of marking-points can be classified into two categories, T-shaped and L-shaped marking-points. As shown in Fig. 2, the pattern of a T-shaped marking-point has a shape like the letter "T", while the pattern of a L-shaped marking-point has a shape like the letter "L".

Orientation. Since the pattern of a T-shaped marking-point is symmetric, we define its orientation as the direction along the axis of symmetry. As for a L-shaped marking-point pattern, we define its orientation as the direction along the marking-line that overlaps with the other marking-line after $90°$ clockwise rotation. Illustrations of the orientations are shown as the red arrows in Fig. 2.

Based on above definitions, a directional marking-point can be represented with a four-dimensional vector:

$$p = \{x, y, s, \theta\} \tag{1}$$

where (x, y) represents the position, s is a binary value representing the shape of the pattern, and θ represents the angular coordinate of pattern's orientation under a polar coordinate system.

3. DMPR-PS: A PARKING-SLOT DETECTION APPROACH BASED ON DIRECTIONAL MARKING-POINT REGRESSION

In this section, our proposed parking-slot detection approach DMPR-PS is presented in detail. As shown in Fig. 3, DMPR-PS comprises two major steps, directional marking-point regression and parking-slot inference.

3.1. Directional marking-point regression

In order to achieve directional marking-point detection, we propose a novel CNN-based multi-attribute regression model with regression objectives specially designed for directional marking-points. Such a regression model is referred to as DMPR in the context.

Given a surround-view image I, DMPR partitions I into a $S \times S$ image grid and extracts a $S \times S \times N$ feature map from I using a CNN. Then during the backpropagation, each N-dimensional vector in the $S \times S \times N$ feature map is assigned to perform the regression of a directional marking-point that falls into the corresponding cell in the $S \times S$ image grid.

In our model, the N-dimensional vector actually consists of 6 elements: cx, cy, s, $\cos\theta$, $\sin\theta$ and confidence C. The confidence predicts the probability of a marking-point falling into that grid cell. (cx, cy) predicts the marking-point's position to the bounds of the grid cell. And s predicts the shape of the directional marking-point. Instead of directly predicting θ, our model predicts two trigonometric value of θ, $\cos\theta$ and $\sin\theta$, since the latter way is more robust in implementation. Based on $\cos\theta$ and $\sin\theta$, θ can be deduced straightforwardly.

With regression objectives defined, the loss function is defined as the sum of squared errors between predictions and ground-truths, and it is expressed as following equation:

$$\begin{aligned} Loss = \sum_{i=1}^{S^2} \{ & (C_i - \hat{C}_i)^2 \\ & + \mathbb{1}_i [(cx_i - \hat{cx}_i)^2 + (cy_i - \hat{cy}_i)^2 + (s_i - \hat{s}_i)^2 \\ & + (\cos\theta_i - \cos\hat{\theta}_i)^2 + (\sin\theta_i - \sin\hat{\theta}_i)^2] \} \end{aligned} \tag{2}$$

The subscript i represents the cell index of the $S \times S$ grid and symbols denoted with $\hat{}$ represent the corresponding ground-truths of the predictions. The operator $\mathbb{1}_i$ denotes whether a marking-point falls into the cell i, which means that we only penalize the marking-point attribute error of the cell i when there is a marking-point falling into that grid cell.

The architecture of the neural network used in DMPR is designed mainly following the advice of current object detection frameworks as well as the common knowledge in this area. Table 1 shows the configurations of this neural network. A bottleneck block with 1×1 squeeze convolution followed by a 3×3 expand convolution is used as the basic building

Table 1. Architecture of the neural network used in DMPR.

Layer Type	Filters	Size / Stride	Output Size (C×H×W)	
conv+norm+relu	32	3×3 / 1	32×512×512	
conv+norm+relu	64	4×4 / 2	64×256×256	
conv+norm+relu	32	1×1 / 1	32×256×256	×1
conv+norm+relu	64	3×3 / 1	64×256×256	
conv+norm+relu	128	4×4 / 2	128×128×128	
conv+norm+relu	64	1×1 / 1	64×128×128	×1
conv+norm+relu	128	3×3 / 1	128×128×128	
conv+norm+relu	256	4×4 / 2	256×64×64	
conv+norm+relu	128	1×1 / 1	128×64×64	×2
conv+norm+relu	256	3×3 / 1	256×64×64	
conv+norm+relu	512	4×4 / 2	512×32×32	
conv+norm+relu	256	1×1 / 1	256×32×32	×2
conv+norm+relu	512	3×3 / 1	512×32×32	
conv+norm+relu	1024	4×4 / 2	1024×16×16	
conv+norm+relu	512	1×1 / 1	512×16×16	×3
conv+norm+relu	1024	3×3 / 1	1024×16×16	
conv+activation	6	1×1 / 1	6×16×16	

block of our network.

The final output tensor with the size of $6 \times 16 \times 16$ is designed according to the requirement of our task. Firstly, as stated above, there are six predictions for each grid cell in the $S \times S$ grid and thus the channel dimension is 6. Secondly, a premise of our regression model is that there could be at most one marking-point falling into a cell in the $S \times S$ grid. If more than one marking-points fall into the same cell, the neural network could not predict both of them in a 6-dimensional vector. Thus, the value of S should be large enough to prevent two marking-points falling into the same cell. S should not be set too large either; otherwise, it will make DMPR computationally expensive. By examining examples in the training set of ps2.0 [20], we set $S = 16$ in our implementation.

3.2. Parking-slot inference

After detecting directional marking-points and applying non-maximum suppression, parking-slots could be inferred from the detected marking-points. The inference procedure comprises two steps, inappropriate marking-point pair filtering and directional marking-point pairing.

Filtering inappropriate marking-point pairs is necessary before the directional marking-point pairing. First, the distance of a pair of marking-points should satisfy a distance constraint. For example, in Fig. 4, P_1 and P_2 are a pair of detected marking-points. But apparently they do not form a valid entrance-line because the space between them is too narrow for a car to park inside it. To exclude such invalid cases, two ranges of the entrance-line distance, which

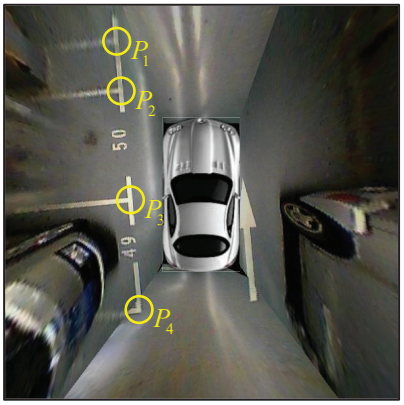


Fig. 4. Illustration of inappropriate marking-point pairs.

respectively correspond to vertical parking-slots and parallel parking-slots, are obtained as a priori knowledge. Then they are used as the distance constraint to filter out marking-point pairs with inappropriate distances. Second, the kind of marking-point pair that passes through a third point needs to be filtered out. In Fig. 4, for marking-points P_2 and P_4, though their orientations and shapes satisfy one of the valid cases of entrance-lines, they do not form a valid entrance-line because they are not adjacent. We could exclude these invalid cases by examining whether there is a third marking-point falling on the entrance-line they form. For marking-points P_2 and P_4, a third point P_3 falls between P_2 and P_4 and thus they cannot form a valid entrance-line.

After filtering out inappropriate marking-point pairs, remaining pairs are sent to determine whether they comply to one of the valid cases of parking-slot entrance-lines. As shown in Fig. 5, for a pair of marking-points (A, B) that form an entrance-line, both marking-points can be classified into 5 cases, as (1) $\sim$ (5) shown in the figure. Then, for these 5 cases of marking-points, there are 16 combinations where two marking-points form a valid entrance-line. These 16 combi-

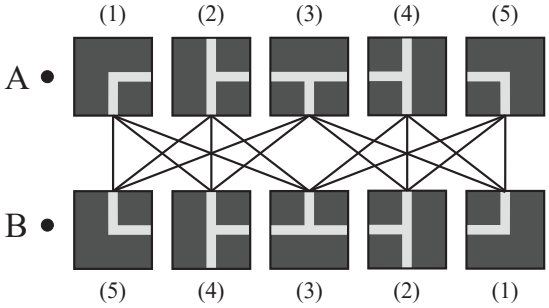


Fig. 5. For a pair of marking-points (A, B) that form an entrance-line, each marking-point can be classified into 5 cases shown as (1) $\sim$ (5). For these five cases of marking-points, there are 16 combinations where these two marking-points form a valid entrance-line.

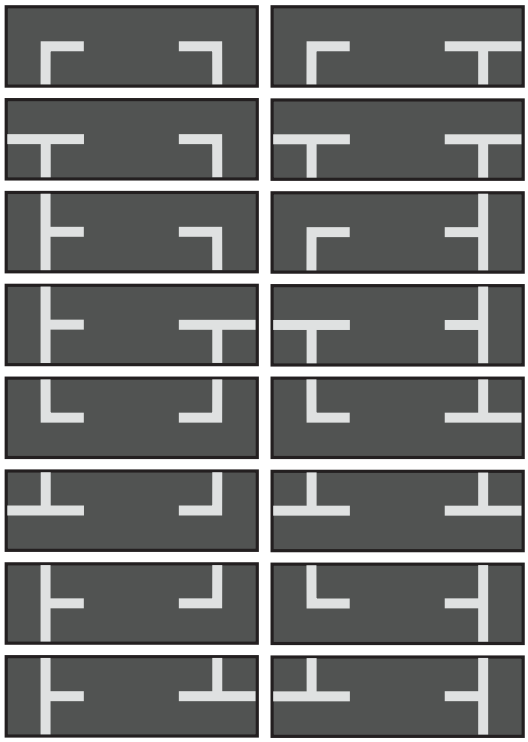


Fig. 6. 16 valid cases of entrance-lines.

nations corresponds to 16 valid cases of entrance-lines shown in Fig. 6. Thus for each marking-point pair, we first determine whether both of the marking-points belong to one of the 5 marking-point cases by comparing shapes and orientations, and then determine whether the combination of these two marking-points matches one of the 16 valid cases of entrance-lines. If both conditions are satisfied, we can deem that these two marking-points form a valid entrance-line and an ordered marking-point pair can be determined. Thus the corresponding parking-slot of this entrance-line can finally be located.

4. EXPERIMENTAL RESULTS

To train DMPR-PS and validate its performance, we used the benchmark dataset ps2.0 [20], which was established for the study of vision-based parking-slot detection. ps2.0 is the largest and the most comprehensive one of its kind. It contains 12,165 surround-view images collected from typical indoor and outdoor parking sites under various illumination conditions. It needs to be noted that shape and orientation information of marking-points was not provided in original ps2.0 and thus we manually label these information in this work.

4.1. Training

Before training, the training set was augmented using image rotation. For each sample, we rotate the image and label 5 degree per time to generate a new sample, until the image is back to its original position. During training, we used Adam optimizer with 10^{-4} as initial learning rate. We trained our network on Nvidia Titan Xp with batch size of 24 for 12 epochs.

4.2. Directional marking-point detection experiment

In this experiment, the evaluation of directional marking-point detection was conducted on the test set. Precision-recall rates were used as the evaluation metric, which are defined as:

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{3}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{4}$$

As defined above, a directional marking-point is represented as $P = \{x, y, s, \theta\}$, while (x, y) represents marking-point's position, s represents the shape of the marking-point pattern, and θ represents the angular coordinate of pattern's orientation in degree.

Suppose that $P_t = \{x_t, y_t, s_t, \theta_t\}$ is a labeled ground-truth directional marking-point and $P_d = \{x_d, y_d, s_d, \theta_d\}$ is a detected one. We defined the following conditions:

$$\|(x_t - x_d, y_t - y_d)\|_2 < 10 \tag{5}$$

$$|\theta_t - \theta_d| < 30^\circ \text{ or } 360^\circ - |\theta_t - \theta_d| < 30^\circ \tag{6}$$

$$s_t = s_d \tag{7}$$

If above conditions are satisfied, we deem that P_t is correctly detected and P_d is a true positive.

With our experimental settings, the number of true positives is 4510, the number of false negatives is 19, and the number of false positives is 20. Accordingly, the precision rate is equal to 99.56% and the recall rate is equal to 99.58%.

4.3. Parking-slot detection experiment

In this experiment, we evaluated the overall performance of various parking-slot detection approaches. The evaluation was conducted on the test set and precision-recall rates are employed as the performance metric. In addition to DMPR-PS, we also evaluated the performance of several representative methods in this field, including Wang *et al.*'s method [10], Hamda *et al.*'s method [11], PSD_L [19], and DeepPS [20].

As described in Sect. 1, the parking-slot detection problem can be formulated as the detection of an ordered marking-point pair. Represent a parking-slot as $S = \{\boldsymbol{o}, \boldsymbol{d}\}$, while $\boldsymbol{o}$ and $\boldsymbol{d}$ represent two points in an ordered marking-point pair. For each labeled ground-truth $S_t = \{\boldsymbol{o_t}, \boldsymbol{d_t}\}$, if there is a detected parking-slot $S_d = \{\boldsymbol{o_d}, \boldsymbol{d_d}\}$ satisfying $\|\boldsymbol{o_t} - \boldsymbol{o_d}\|_2 < 10$ and $\|\boldsymbol{d_t} - \boldsymbol{d_d}\|_2 < 10$, we deem that S_t is correctly detected and S_d is a true positive.

We adjusted the parameters of all the competing methods to make their precision rates greater than 98% on the test set.

Table 2. Performance comparison of parking-slot detection.

method	precision	recall
Wang *et al.*s method [10]	98.29%	58.33%
Hamda *et al.*s method [11]	98.45%	61.37%
PSD_L [19]	98.41%	86.96%
DeepPS [20]	98.99%	99.13%
DMPR-PS	99.42%	99.37%

The results are summarized in Table 2. From Table 2, it can be observed that when operating at a high precision rate, our method outperforms all the other methods with a precision rate of 99.42% and a recall rate of 99.37%.

We also measured the speed of our method implemented with PyTorch on Nvidia Titan Xp. The average time for our method to process one image frame is about 12 ms. It is about 30% faster than another DCNN-based approach DeepPS, which is about 17 ms tested on Nvidia Titan Xp.

5. CONCLUSION

In this paper, we propose a parking-slot detection method based on directional marking-point regression. A key feature of our method is a regression model specially designed for the parking-slot detection. This model allows us to predict marking-points and their surrounding patterns simultaneously. Experiments have proven that our method outperforms other state-of-the-art methods, while also achieving faster detection speed comparing with a state-of-the-art DCNN-based approach DeepPS. In the future we will continue devoting our efforts to regression-based parking-slot detection approaches, in searching for a more efficient and robust solution.

6. ACKNOWLEDGEMENT

This research was funded in part by the Natural Science Foundation of China under Grant No. 61672380, in part by the National Key Research and Development Project under Grant No. 2017YFE0119300, and in part by the Fundamental Research Funds for the Central Universities under Grant No. 2100219068.

7. REFERENCES

[1] W. J. Park, B. S. Kim, D. E. Seo, D. S. Kim, and K. H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *IEEE Intell. Veh. Symp.*, 2008, pp. 1039–1044.

[2] S. H. Jeong, C. G. Choi, J. N. Oh, P. J. Yoon, B. S. Kim, M. Kim, and K. H. Lee, "Low cost design of parallel parking assist system based on an ultrasonic sensor," *Int. J. Autom. Technol.*, vol. 11, no. 3, pp. 409–416, 2010.

[3] J. Zhou, L. E. Navarro-Serment, and M. Hebert, "Detection of parking spots using 2D range data," in *IEEE Int. Conf. Intell. Transp. Syst.*, 2012, pp. 1280–1287.

[4] R. Dub, M. Hahn, M. Schtz, J. Dickmann, and D. Gingras, "Detection of parked vehicles from a radar based occupancy grid," in *IEEE Intell. Veh. Symp.*, 2014, pp. 1415–1420.

[5] A. Loeffler, J. Ronczka, and T. Fechner, "Parking lot measurement with 24 GHz short range automotive radar," in *IEEE Int. Radar Symp.*, 2015, pp. 137–142.

[6] H. G. Jung, D. S. Kim, P. J. Yoon, and J. Kim, "Structure analysis based parking slot marking recognition for semi-automatic parking system," in *IAPR Int. Workshop Struct. Syntact. Patt. Recog.*, 2006, pp. 384–393.

[7] H. G. Jung, Y. H. Lee, and J. Kim, "Uniform user interface for semiautomatic parking slot marking recognition," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 616–626, 2010.

[8] J. Xu, G. Chen, and M. Xie, "Vision-guided automatic parking for smart car," in *IEEE Intell. Veh. Symp.*, 2000, pp. 725–730.

[9] H. G. Jung, D. S. Kim, P. J. Yoon, and J. Kim, "Parking slot markings recognition for automatic parking assist system," in *IEEE Intell. Veh. Symp.*, 2006, pp. 106–113.

[10] C. Wang, H. Zhang, M. Yang, X. Wang, L. Ye, and C. Guo, "Automatic parking based on a bird's eye view vision system," *Adv. Mech. Eng.*, vol. 6, pp. 847406:1–13, 2014.

[11] K. Hamada, Z. Hu, M. Fan, and H. Chen, "Surround view based parking lot detection and tracking," in *IEEE Intell. Veh. Symp.*, 2015, pp. 1106–1111.

[12] S. Lee and S. Seo, "Available parking slot recognition based on slot context analysis," *IET Intell. Transp. Syst.*, vol. 10, no. 9, pp. 594–604, 2016.

[13] J. K. Suhr and H. G. Jung, "Automatic parking space detection and tracking for underground and indoor environments," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5687–5698, 2016.

[14] J. K. Suhr and H. G. Jung, "A universal vacant parking slot recognition system using sensors mounted on off-the-shelf vehicles," *Sensors*, vol. 18, no. 4, pp. 1213, 2018.

[15] Q. Li, C. Lin, and Y. Zhao, "Geometric features-based parking slot detection," *Sensors*, vol. 18, no. 9, pp. 2821, 2018.

[16] C. Jang, C. Kim, S. Lee, S. Kim, S. Lee, and M. Sunwoo, "Replannable automated parking system with a standalone around view monitor for narrow parking lots," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–14, 2019.

[17] J. K. Suhr and H. G. Jung, "Full-automatic recognition of various parking slot markings using a hierarchical tree structure," *Opt. Eng.*, vol. 52, no. 3, pp. 037203:114, 2013.

[18] J. K. Suhr and H. G. Jung, "Sensor fusion-based vacant parking slot detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 21–36, 2014.

[19] L. Li, L. Zhang, X. Li, X. Liu, Y. Shen, and L. Xiong, "Vision-based parking-slot detection: A benchmark and a learning-based approach," in *ICME*, 2017, pp. 649–654.

[20] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset," *IEEE Trans. IP*, vol. 27, no. 11, pp. 5350–5364, 2018.